Amendments to the Claims

The listing of claims will replace all prior versions, and listings of claims in the application.

1. (*currently amended*) A method for optimizing the transmission of TCP/IP traffic between a cable modem and a cable modem termination system (CMTS) in a DOCSIS network that supports a dynamic delta encoding header suppression protocol, comprising the steps of:

~~(a) determining whether the CMTS supports a dynamic delta encoding header suppression protocol; and~~

~~(b) if the CMTS does support the dynamic delta encoding header suppression protocol, then:~~

(i) transmitting fields in a first protocol header of a first TCP protocol packet from the cable modem,

(ii) suppressing a redundant field in a second protocol header of a subsequent TCP protocol packet, and

(iii) transmitting a delta-encoded value for each non-redundant field in said second protocol header of said subsequent TCP protocol packet, wherein said delta-encoded value represents a change in value from a respective non-redundant field in said first protocol header of said first TCP protocol packet.

2. (*previously presented*) The method of claim 1, wherein step (i) further comprises the step of transmitting said first TCP protocol packet with an indicator, wherein said indicator indicates that said first TCP protocol packet is to be learned.

3. (*previously presented*) The method of claim 1, wherein step (i) further comprises the step of transmitting said first TCP protocol packet in its entirety and transmitting said subsequent protocol header in a compressed format.

4. (*previously presented*) The method of claim 1, wherein said subsequent TCP protocol packet includes a bitmapped change byte, wherein bits in said bitmapped change byte indicate at least one non-redundant field in said second protocol header that has said delta encoded value.

5. (*previously presented*) The method of claim 1, further comprising the steps of:

(iv) enabling a receiver to learn said first TCP protocol packet,

(v) enabling a receiver to restore said suppressed redundant field in said second protocol header of said subsequent TCP protocol packet using said first TCP protocol packet,

(vi) enabling a receiver to restore a non-redundant field in said second protocol header of said subsequent TCP protocol packet using said respective delta-encoded value, and

(vii) enabling a receiver to provide said restored second protocol header in front of corresponding received data for transmission over an Internet Protocol network.

6. (*previously presented*) The method of claim 4, further comprising the steps of:

(iv) enabling a receiver to read said bitmapped change byte,

(v) enabling a receiver to retrieve said delta encoded value using said bitmapped change byte,

(vi) enabling a receiver to update said respective non-redundant field in said second protocol header using said delta-encoded value, and

(vii) enabling a receiver to restore said second protocol header to its original format.

7. (*previously presented*) The method of claim 6, further comprising the step of providing said restored second protocol header in front of corresponding received data for transmission over an Internet Protocol network.

8. (*currently amended*) A method for receiving packets by a cable modem termination system (CMTS) from a cable modem in a DOCSIS network, comprising the steps of:

~~(a) receiving a message from the cable modem indicating support for a dynamic delta encoding header suppression protocol; and~~

~~(b) if the CMTS supports the dynamic delta encoding header suppression protocol, then:~~

(i) receiving fields in a first protocol header of a first TCP protocol packet from the cable modem,

(ii) receiving an indication that a redundant field in a second protocol header of a subsequent TCP protocol packet is suppressed, and

(iii) receiving a delta-encoded value for each non-redundant field in said second protocol header of said subsequent TCP protocol packet, wherein said delta-encoded value represents a change in value from a respective non-redundant field in said first protocol header of said first TCP protocol packet.

9. (*previously presented*) The method of claim 8, wherein step (i) further comprises the step of receiving an indicator with said first TCP protocol packet, wherein said indicator indicates that said first TCP protocol packet is to be learned.

10. (*previously presented*) The method of claim 8, wherein said subsequent TCP protocol packet includes a bitmapped change byte, wherein bits in said bitmapped change byte indicate at least one non-redundant field in said second protocol header that has said delta encoded value.

11. (*currently amended*) The method of claim 8, further comprising the steps of:

~~(d)~~ (iv) learning said first TCP protocol packet;

~~(e)~~ (v) using learned information from said first TCP protocol packet to reconstruct said suppressed field in said second protocol header of said subsequent TCP protocol packet; and

~~(f)~~ (vi) using said first TCP protocol packet to reconstruct a non-redundant field in said second protocol header of said subsequent TCP protocol packet.

12. (*previously presented*) The method of claim 11, further comprising the step of restoring said subsequent TCP protocol packet to its original format and transmitting said subsequent TCP protocol packet over an Internet Protocol network.

13. (*currently amended*) A computer program product comprising a computer useable medium including control logic stored therein, said control logic for optimizing the transmission of TCP/IP traffic between a cable modem and a cable modem termination system (CMTS) in a DOCSIS network, said control logic ~~comprising~~ enabling a processor to:

~~first means for enabling a processor to determine whether the CMTS supports a dynamic delta encoding header suppression protocol; and~~

~~second means for enabling a processor, if the CMTS does support the dynamic delta encoding header suppression protocol, to~~

transmit fields in a first protocol header of a first TCP protocol packet,

suppress a redundant field in a second protocol header of a subsequent TCP protocol packet, and

transmit a delta-encoded value for each non-redundant field in said second protocol header of said subsequent TCP protocol packet, wherein said delta-encoded value represents a change in value from a respective non-redundant field in said first protocol header of said first TCP protocol packet.

14. (*currently amended*) The computer program product of claim 13, ~~wherein said second means further comprises means for enabling~~ further including control logic that enables a processor to transmit said first TCP protocol packet with an indicator, wherein said indicator indicates that said first TCP protocol packet is to be learned.

15. (*currently amended*) The computer program product of claim 13, ~~wherein said second means further comprises means for enabling~~ further including control logic that enables a processor to transmit said first TCP protocol packet in its entirety and transmit said subsequent protocol header in a compressed format.

16. (*previously presented*) The computer program product of claim 13, wherein said subsequent TCP protocol packet includes a bitmapped change byte, wherein bits in said bitmapped change byte indicate at least one non-redundant field in said second protocol header that has said delta encoded value.

17. (*currently amended*) The computer program product of claim 13, further ~~comprising~~ including control logic that enables a processor to enable a receiver to:

~~means for enabling a processor to enable a receiver to~~ learn said first TCP protocol packet;

~~means for enabling a processor to enable a receiver to~~ restore said suppressed redundant field in said second protocol header of said subsequent TCP protocol packet using said first TCP protocol packet;

~~means for enabling a processor to enable a receiver to~~ restore a non-redundant field in said second protocol header of said subsequent TCP protocol packet using said respective delta-encoded value; and

~~means for enabling a processor to enable a receiver to~~ provide said restored second protocol header in front of corresponding received data for transmission over an Internet Protocol network.

18. (*currently amended*) The computer program product of claim 16, further ~~comprising~~ including control logic that enables a processor to enable a receiver to:

~~means for enabling a processor to enable a receiver to~~ read said bitmapped change byte,

~~means for enabling a processor to enable a receiver to~~ retrieve said delta encoded value using said bitmapped change byte,

~~means for enabling a processor to enable a receiver to~~ update said non-redundant field in said second protocol header using said delta-encoded value, and

~~means for enabling a processor to enable a receiver to~~ restore said second protocol header to its original format.

19. (*currently amended*) The computer program product of claim 18, further ~~comprising means for enabling~~ including control logic that enables a processor to provide said restored second protocol header in front of corresponding received data for transmission over an Internet Protocol network.

20. (*currently amended*) A computer program product comprising a computer useable medium including control logic stored therein, said control logic for enabling packets to be received by a cable modem termination system (CMTS) from a cable modem in a DOCSIS network, said control logic ~~comprising~~ enabling a processor to:

~~first means for enabling a processor to receive a message from the cable modem indicating support for a dynamic delta encoding header suppression protocol; and~~

~~second means for enabling a processor, if the CMTS supports the dynamic delta encoding header suppression protocol, to~~ receive fields in a first protocol header of a first TCP protocol packet from the cable modem,

receive an indication that a redundant field in a second protocol header of a subsequent TCP protocol packet is suppressed, and

receive a delta-encoded value for each non-redundant field in said second protocol header of said subsequent TCP protocol packet, wherein said delta-encoded value represents a change in value from a respective non-redundant field in said first protocol header of said first TCP protocol packet.

21. (*currently amended*) The computer program product of claim 20, ~~wherein said second means further comprises means for enabling~~ further including control logic that enables a processor to receive an indicator with said first TCP protocol packet, wherein said indicator indicates that said first TCP protocol packet is to be learned.

22. (*previously presented*) The computer program product of claim 20, wherein said subsequent TCP protocol packet includes a bitmapped change byte, wherein bits in said bitmapped change byte indicate at least one non-redundant field in said second protocol header that has said delta encoded value.

23. (*currently amended*) The computer program product of claim 20, further ~~comprising~~ including control logic that enables a processor to:

~~means for enabling a processor to~~ learn said first TCP protocol packet;

~~means for enabling a processor to~~ use learned information from said first TCP protocol packet to reconstruct said suppressed field in said second protocol header of said subsequent TCP protocol packet; and

~~means for enabling a processor to~~ use said first TCP protocol packet to reconstruct a non-redundant field in said second protocol header of said subsequent TCP protocol packet.

24. (*currently amended*) The computer program product of claim 23, further ~~comprising means for enabling~~ including control logic that enables a processor to restore said subsequent TCP protocol packet to its original format and transmit said subsequent TCP protocol packet over an Internet Protocol network.